Keyword Search on Graph-Structured Data

S. Sudarshan
CSE Dept, IIT Bombay

Joint work with
Prof. Soumen Chakrabarti,
Gaurav Bhalotia, Charuta Nakhe, Arvind Hulgeri,
Varun Kacholia, Shashank Pandit, Rushi Desai, Hrishi K.,
Bhavana Dalvi and Meghana Kshirsagar, et al.

Jan 2009
Keyword Search on Semi-Structured Data

- Keyword search of documents on the Web has been enormously successful
- Much data is resident in databases
  - Organizational, government, scientific, medical data
  - Deep web
- Goal (circa 2002/3): keyword querying of data from
  - relational databases
  - multiple data sources, with different data models
    - Often with no schema or partially defined schema
- Extra goals (circa 2009):  
  - Web search: from documents to entities  
    - e.g. Google squared
Keyword Search on Structured/Semi-Structured Data

Key differences from IR/Web Search:
- Normalization (implicit/explicit) splits related data across multiple tuples
- To answer a keyword query we need to find a (closely) connected set of entities that together match all given keywords
  - soumen crawling or soumen byron

BANKS: Keyword search...
Focused Crawling ...

Sudarshan
Soumen C.
Byron Dom

paper
writes
author
Graph Data Model

- Lowest common denominator across many data models
  - Relational
  - XML
  - HTML
  - Documents
  - Knowledge representation
  - Social network and other network data
Query/ Answer Models

- Basic query model:
  - Keywords match node text/labels

- Two answer models
  - tree that connects nodes matching query keywords
  - nodes in proximity to (near) query keywords

Eg. "Soumen Byron"
Answer Ranking for Connection Queries

- Naïve model: answer trees ranked by number of edges

- Problem:
  - Some tuples are connected to many other tuples
    - E.g. highly cited papers, popular web sites
  - Highly connected tuples create misleading shortcuts
    - six degrees of separation

- Solution: use directed edges with edge weights
  - and rank answers by directed edge weights from root to leaves of answer tree
    - details in BANKS paper in ICDE 2002
Answer Ranking (Cont.)

- **Node prestige:**
  - More incoming edges $\rightarrow$ higher prestige
  - Google PageRank style transfer of prestige
    - Node weight computing using biased random walk model
  - Plus standard IR techniques such as TF/IDF

- **Overall score of answer tree A**
  - combine tree and node scores
    - details in BANKS papers in ICDE 2002 and VLDB 2005
Anecdotal results on DBLP Bibliography

- “Transaction”: Jim Gray’s classic paper and textbook at the top because of prestige (# of citations)
- “soumen sudarshan”: several coauthored papers, links via common co-authors
Answer Models

- Tree Answer Model
- Proximity (near query) model
Proximity Queries

- Node weight by proximity
  - author (near OLAP) (on DBLP)
  - faculty (near earthquake) (on IITB thesis database)
- Node prestige > if close to multiple nodes matching near keywords

Example applications
- Finding experts on a particular area
  - OLAP over uncertain..
  - Computing sparse cubes...
  - Widom
  - Allocation in OLAP...
  - Overview of OLAP...
Proximity via Spreading Activation

Idea:
- Each “near” keyword has activation of 1
  - Divided among nodes matching keyword, proportional to their node prestige
- Each node
  - keeps fraction $1-\mu$ of its received activation and
  - spreads fraction $\mu$ amongst its neighbors
- Combine activation $a_i$ received from neighbors
  - $a = 1 - \prod(1-a_i)$ (belief function)
- Graph may have cycles
  - Iterate till convergence
Example Answers

- Anecdotal results on DBLP Bibliography
  - author (near recovery): Dave Lomet, C. Mohan, etc
  - sudarshan (near change): Sudarshan Chawate
  - sudarshan(near query): S. Sudarshan

- And on IITB Thesis database:
  - faculty (near earthquake): Jangid, Ravi Sinha, P. Banerji, ..
Related Work

- Keyword querying on relational databases
  - DBExplorer (Microsoft, ICDE02) Discover (UCSD, VLDB02, VLDB03), Mragyati (DASFAA03)
  - Use SQL generation, not applicable to arbitrary graphs
  - ranking based only on #nodes/edges

- Proximity Search
  - Goldman, et al. [VLDB98]
  - Object Rank [VLDB04]
Finding Answers using Backward Expanding Search

Query: soumen byron

paper

writes

authors

Focused Crawling

Soumen C.

Byron Dom
Backward Expanding Search

Backward Expanding Search Algorithm (Bhalotia et al, ICDE02):

- Intuition: find vertices from which a forward path exists to at least one node from each Si.
- Run **concurrent single source shortest path** algorithm from each node matching a keyword
  - Create an iterator for each node matching a keyword
  - Do best-first search across iterators
  - Output an answer when its root has been reached from each keyword
    - Answer heap to collect and output results in score order
Bidirectional Search: Motivation

Several issues in efficient implementation
- details in VLDB05 paper

Backward search

How about searching in forward direction?

Backward search doesn’t seem useful because of so many keyword nodes
Performance Results

- Two versions of backward search:
  - Iterator per node (MI-Bkwd) vs Iterator per keyword (SI-Bkwd)
  - Origin size: number of nodes matching keywords

![Graphs showing time ratio for different keyword counts and origin sizes.](image)
External Memory Graph Search

- Problem: what if graph size > memory?
  - Alternative 1: Virtual Memory
    - thrashing
  - Alternative 2 (for relational data): SQL
    - not good for top-K answer generation across multiple SQL queries
  - Alternative 3: use graph clustering to compress graph, search on compressed graph
    - Problem: how to get correct top-K answers?
    - New idea: Multi-granular graph representation, with incremental expansion during search
    - External memory BANKS [Dalvi et al, VLDB 2008]
Conclusions

- Keyword search on graphs continues to grow in importance
  - E.g. graph representation of extracted knowledge
  - Annotating Web pages with entities
  - Entity search instead of Web page search

- Ongoing/Future Work
  - Integration with existing applications
    - To provide more natural display of results, hiding schema details
    - Authorization
  - Graph search in a parallel cluster
  - Goal: search integrated WWW/Wikipedia graph
  - New search algorithms
Thanks!
BANKS References

- **Keyword Searching and Browsing in databases using BANKS**, Gaurav Bhalotia, Arvind Hulgeri, Charuta Nakhe, Soumen Chakrabarti, S. Sudarshan
  ICDE 2002

- **User Interaction in the BANKS System**, Demo paper, B. Aditya, Soumen Chakrabarti, Rushi Desai, Arvind Hulgeri, Hrishikesh Karambelkar, Rupesh Nasre, Parag, S. Sudarshan
  ICDE 2003

- **Bidirectional Expansion For Keyword Search on Graph Databases**, Varun Kacholia, Shashank Pandit, Soumen Chakrabarti, S Sudarshan, Rushi Desai and Hrishikesh Karambelkar,
  VLDB 2005

- **Keyword Search on External Memory Data Graphs**
  Bhavana Dalvi, Meghana Kshirsagar and S. Sudarshan,
  VLDB 2008
The BANKS System

- Available on the web, with DBLP, IMDB and IITB ETD data
  - [http://www.cse.iitb.ac.in/banks/](http://www.cse.iitb.ac.in/banks/)
- No programming needed for customization
  - Minimal preprocessing to create indices and give weights to links
- Provides keyword search coupled with extensive browsing features
  - Schema browsing + data browsing
  - Hyperlinks are automatically added to all displayed results
  - Browsing data by grouping and creating crosstabs
  - Graphical display of data: bar charts, pie charts, etc